



Fig1 Standalone.vsd

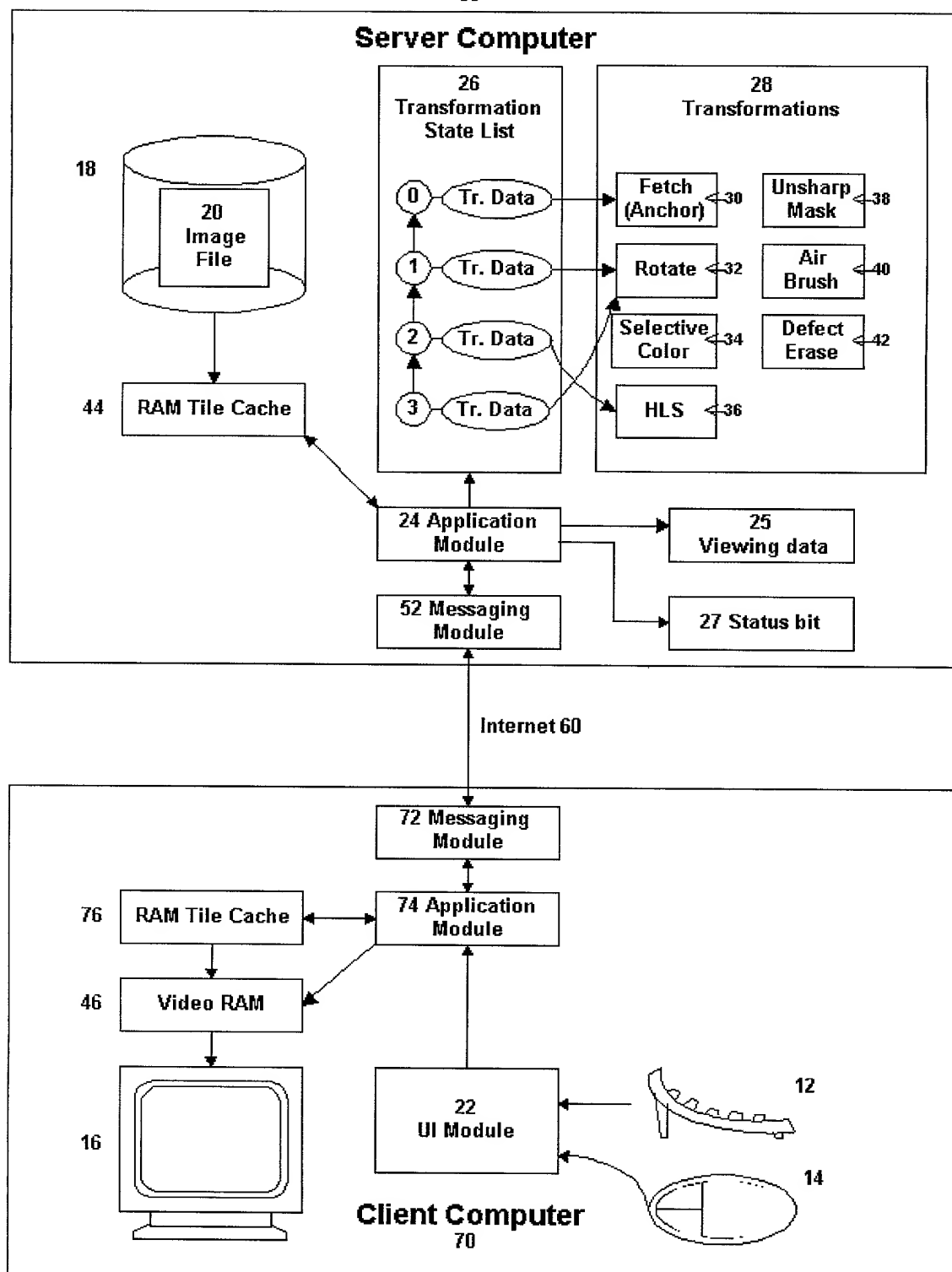
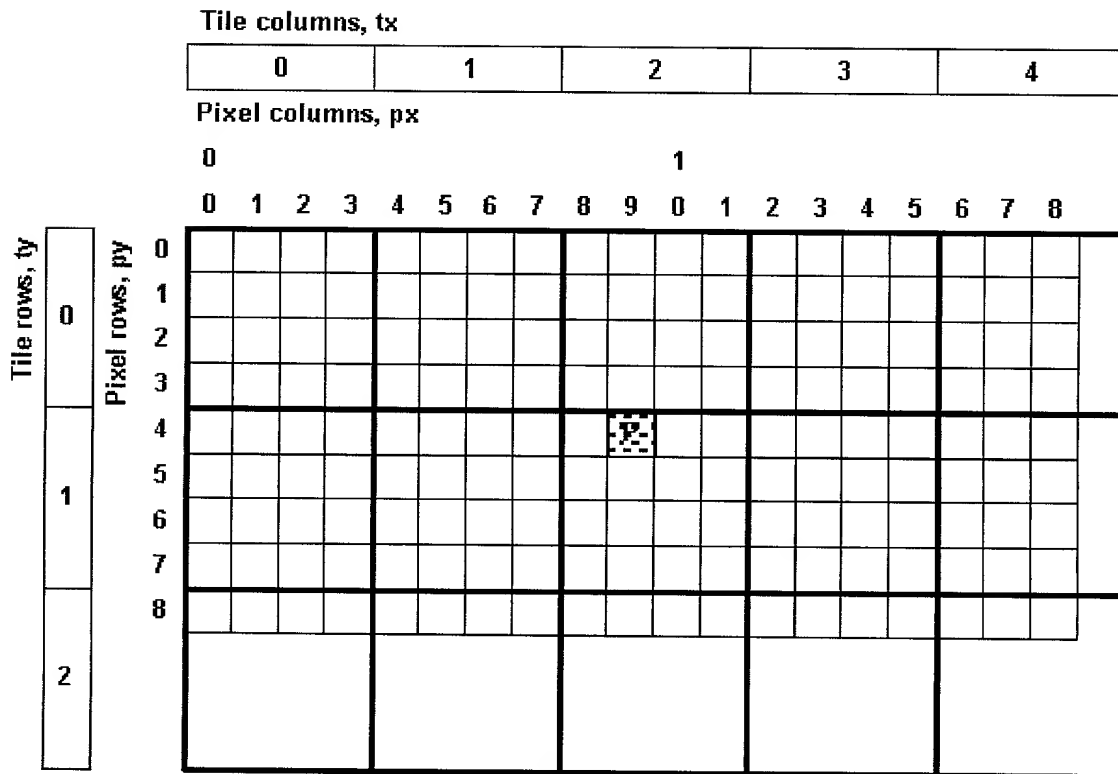
**Figure 2**

Fig2Network vsd



Pixel-Tile Conversion schema

- (1) all tiles are of dimension n by n ;
- (2) the image origin is at the upper left;
- (3) the coordinates of the pixel at the origin are $(x=0, y=0)$;
- (3) pixels are identified by their (column, row) pair, (x, y) , offset from the origin;
- (4) the pixel at the top left of the top-left tile is the origin pixel;
- (5) tiles are identified by their (column, row) pair, (x, y) , offset from the origin;
- (6) $n = 4$;

Then:

the pixel P with coordinates (px, py) is in the tile T with coordinates $(tx, ty) = (px/n, py/n)$, where the symbol $"/"$ denotes integer division with the remainder discarded.

The coordinates (rx, ry) of pixel P relative to the origin of tile T are $(rx, ry) = (px \% n, py \% n)$, where the symbol $"\%"$ denotes the remainder after integer division.

In the example shown, P has coordinates $(px, py) = (9, 4)$; hence it lies in tile T with coordinates $(tx, ty) = (2, 1)$, and its coordinates relative to the origin of that tile are $(rx, ry) = (1, 0)$.

Figure 3

PixelTile.vsd

24 Application Module

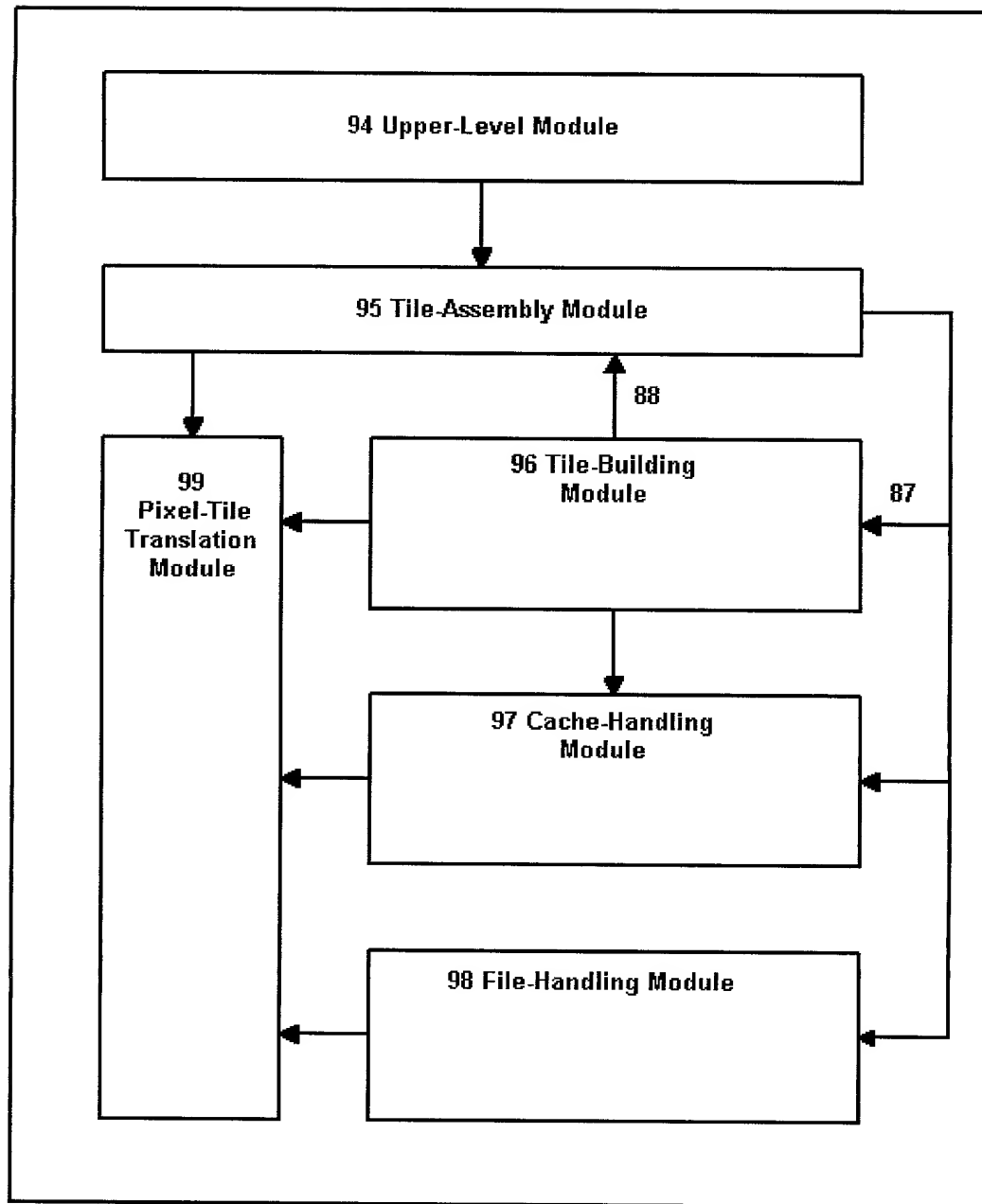


Figure 4
OverHier.vsd

Figure 1 consists of 12 scatter plots, labeled (a) through (l), arranged in a 6x2 grid. Each plot shows the relationship between the 'Number of children' (x-axis, ranging from 0 to 10) and a specific variable (y-axis). The variables are: (a) age, (b) sex, (c) marital status, (d) education, (e) income, (f) employment, (g) housing, (h) health, (i) social capital, (j) life satisfaction, (k) mental health, and (l) physical health. In all plots, there is a clear positive correlation, with the y-axis values increasing as the number of children increases. The y-axis scales vary between plots, reflecting the different ranges of the variables.

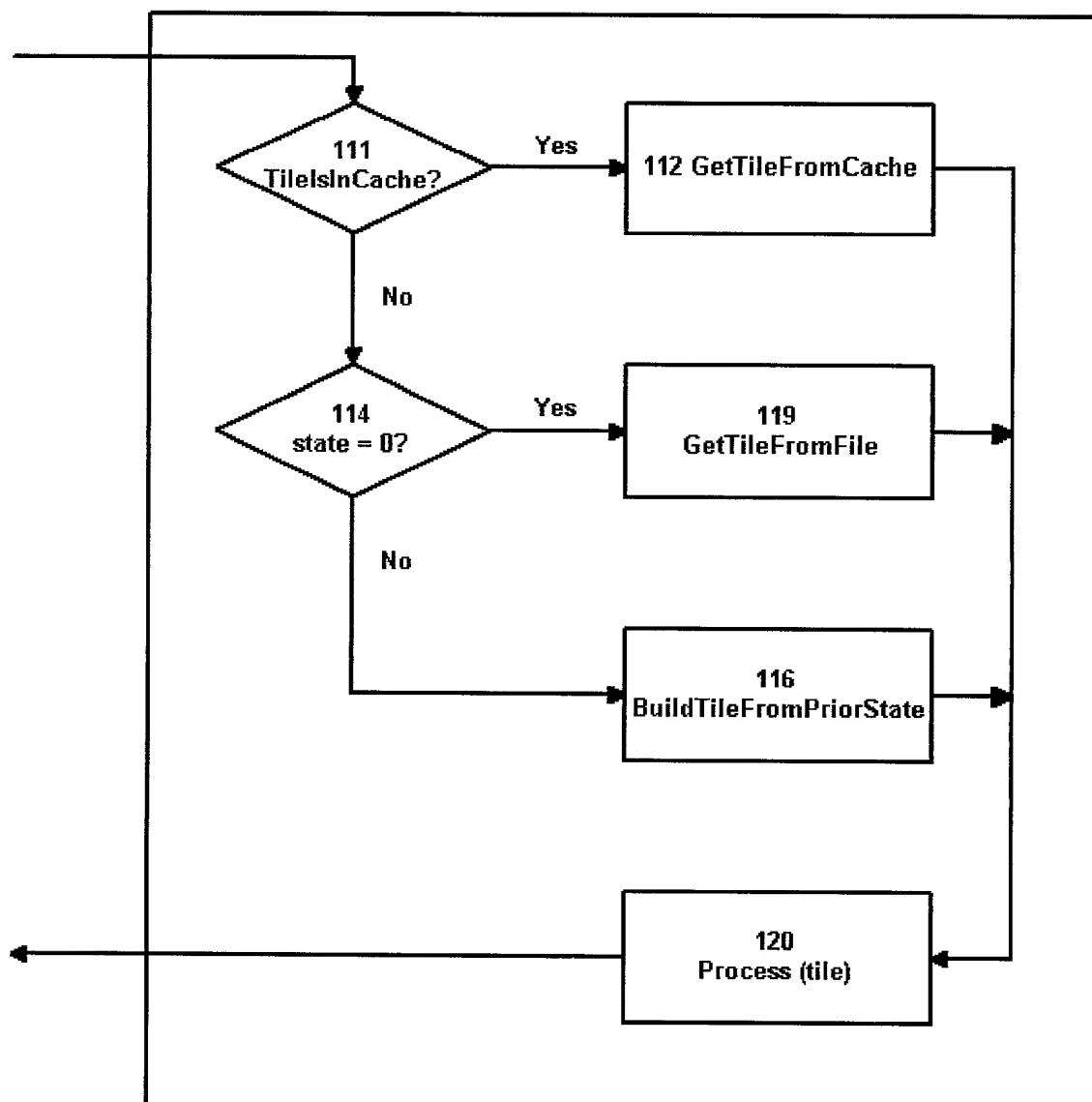


Figure 5
FlowBasic.vsd


```

141 // Figure 6e
142 UnlockTiles (state, scale, region)
143 {
144     ListOfTilesToGet = FindAllTilesIntersecting (state, scale, region);
145     For (all tiles in ListOfTilesToGet)
146         If (TileIsInCache (state, scale, x, y))
147             UnlockTile (state, scale, x, y);
148 }
149
150 // Figure 6f
151 CacheTile (tile)
152 {
153     If (!RoomInCache())
154         PurgeTilesFromCache();
155     AddTileToCache (tile);
156 }

```

Figure 6, parts e-f

Fig6Acode.vsd

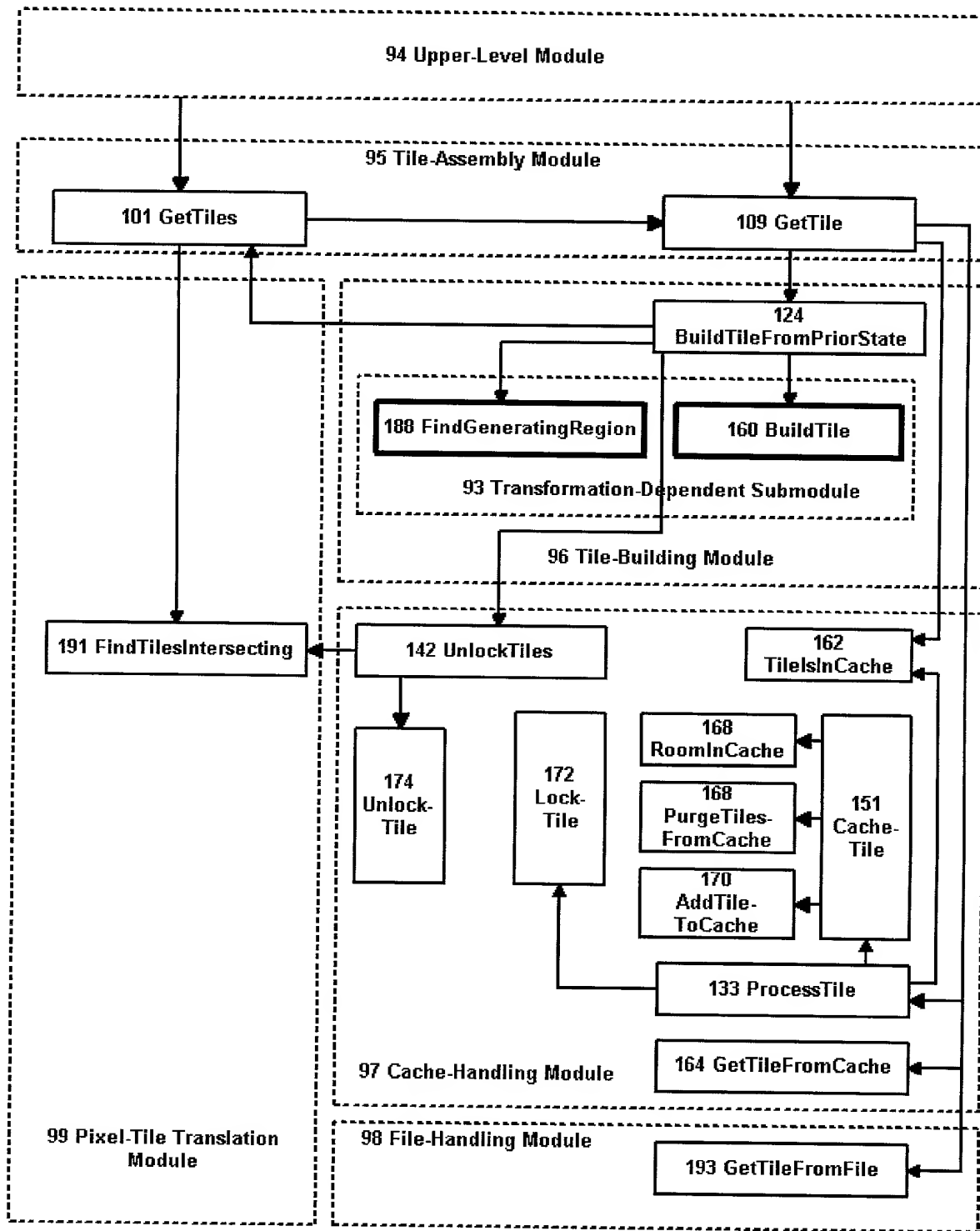
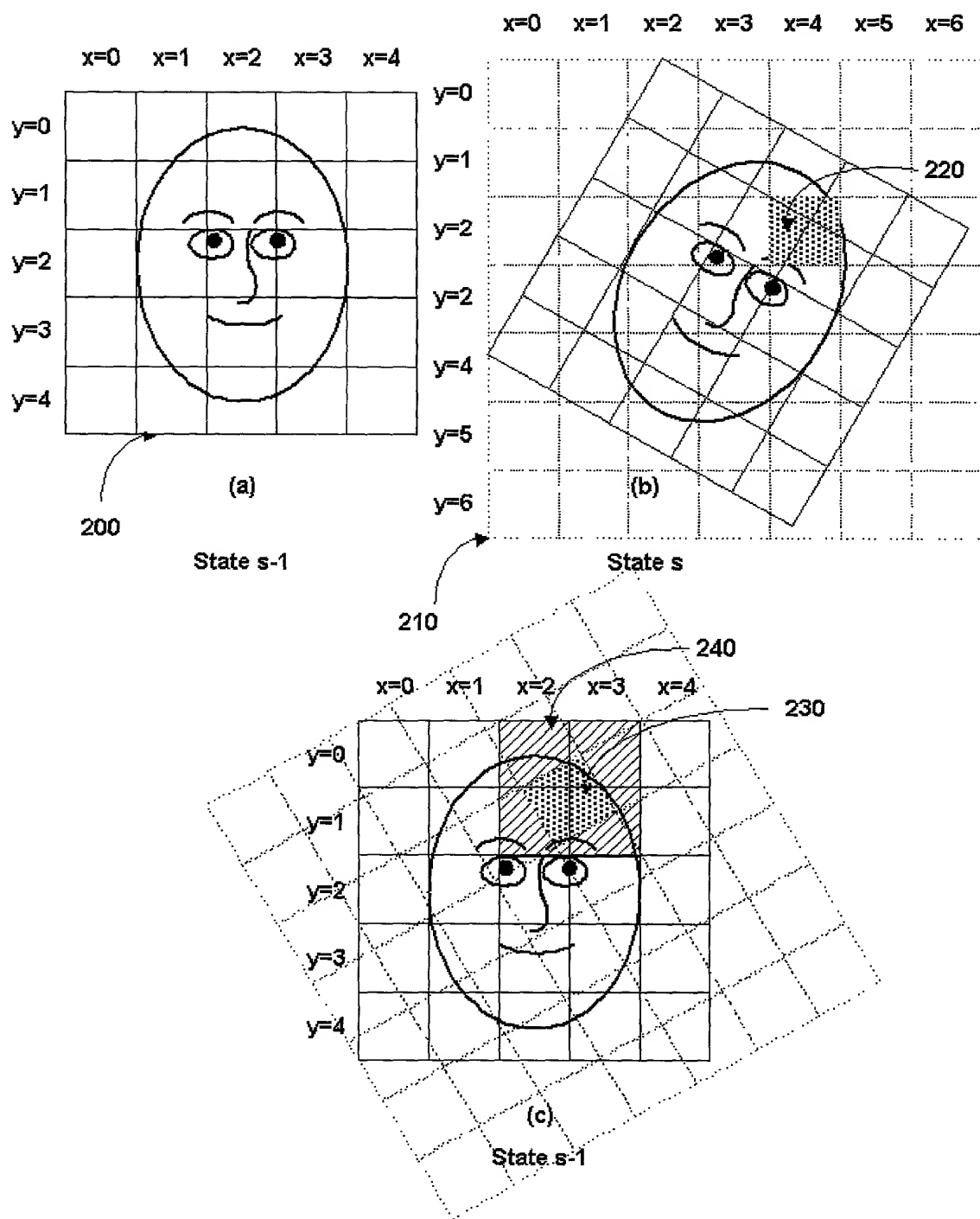


Figure 7
Funchier.yxd



Inductive Image Generation

Figure 8
Rotate.vsd